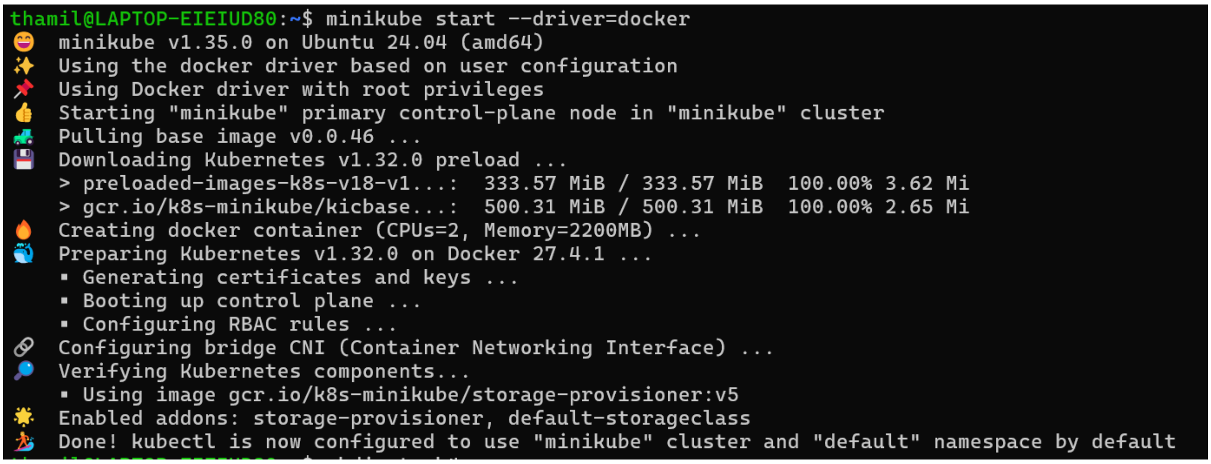
TASK 4 – Kubernet Using Shell Script

Step 1: MiniKube

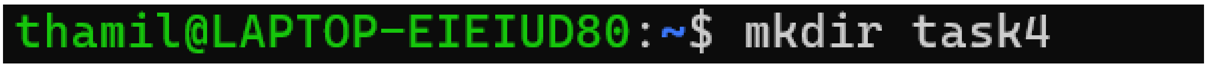
Start the minikube using minikube start command



```
thamil@LAPTOP-EIEIUD80:~$ minikube start --driver=docker
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on user configuration
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
💾  Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 3.62 Mi
    > gcr.io/k8s-minikube/kicbase...:  500.31 MiB / 500.31 MiB  100.00% 2.65 Mi
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Step 2: Folder Creation

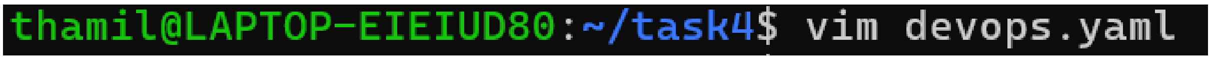
Create a folder named task4



```
thamil@LAPTOP-EIEIUD80:~$ mkdir task4
```

Step 3: New Yaml File

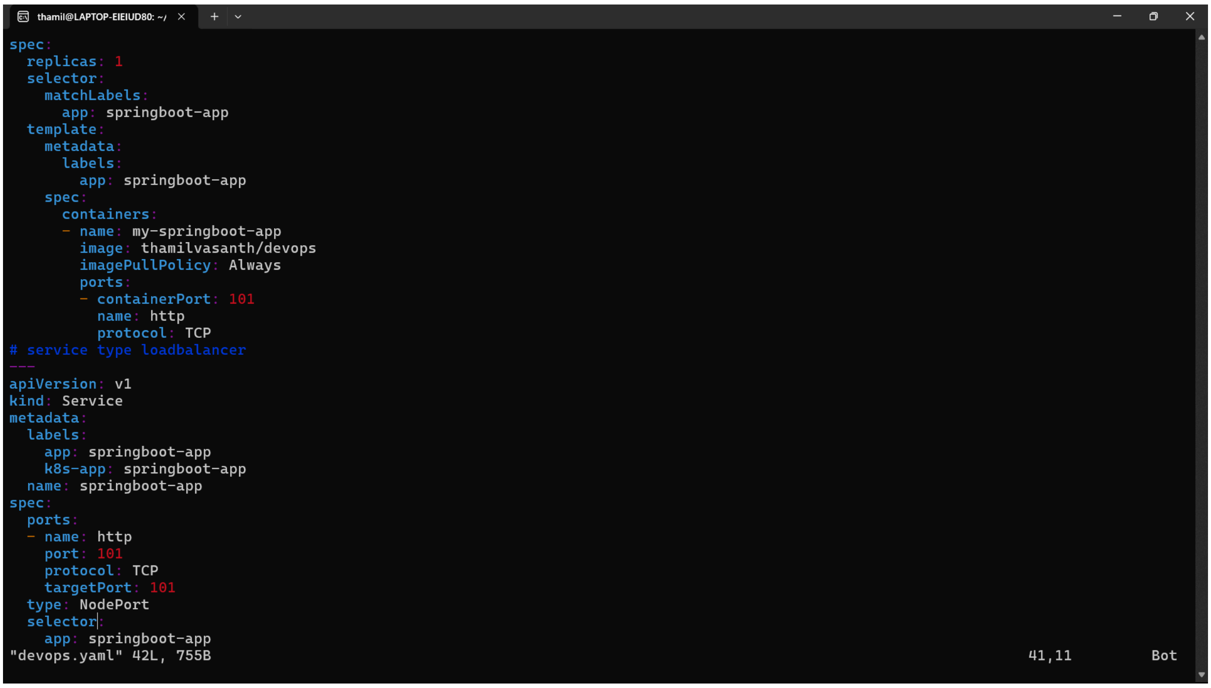
Create a new vim file named devops.yaml



```
thamil@LAPTOP-EIEIUD80:~/task4$ vim devops.yaml
```

Step 4: Yaml file

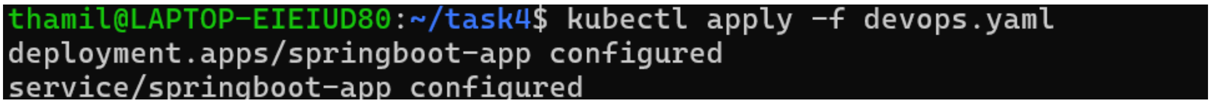
Enter the yaml file code using the insert



```
spec:
  replicas: 1
  selector:
    matchLabels:
      app: springboot-app
  template:
    metadata:
      labels:
        app: springboot-app
    spec:
      containers:
      - name: my-springboot-app
        image: thamilvasanth/devops
        imagePullPolicy: Always
        ports:
        - containerPort: 101
          name: http
          protocol: TCP
# service type loadbalancer
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: springboot-app
    k8s-app: springboot-app
  name: springboot-app
spec:
  ports:
  - name: http
    port: 101
    protocol: TCP
    targetPort: 101
  type: NodePort
  selector:
    app: springboot-app
"devops.yaml" 42L, 755B                                                              41,11         Bot
```

Step 5: Apply

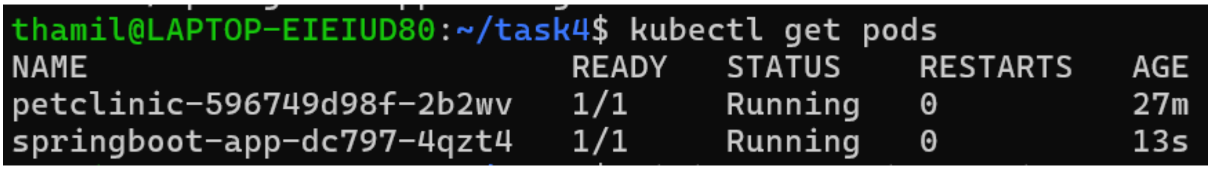
Apply the changes made in the devops.yaml file



```
thamil@LAPTOP-EIEIUD80:~/task4$ kubectl apply -f devops.yaml
deployment.apps/springboot-app configured
service/springboot-app configured
```

Step 6: Get Pods

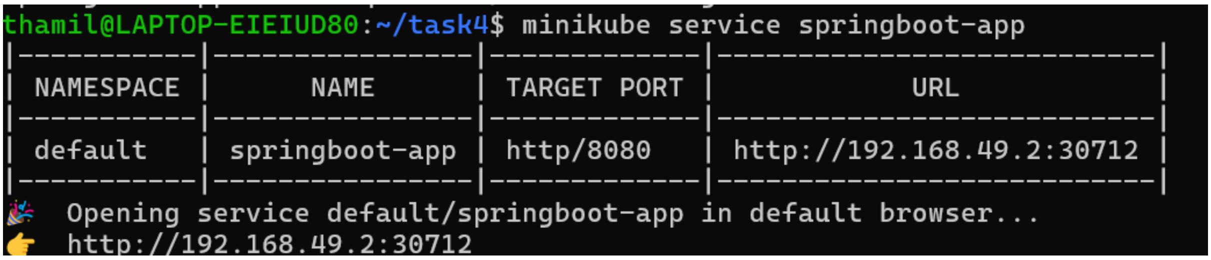
Get the pods information to check if it is running or not.



```
thamil@LAPTOP-EIEIUD80:~/task4$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
petclinic-596749d98f-2b2wv        1/1     Running   0          27m
springboot-app-dc797-4qzt4        1/1     Running   0          13s
```

Step 7: Service

Open the service springboot-app in the browser



```
thamil@LAPTOP-EIEIUD80:~/task4$ minikube service springboot-app
|-----------|----------------|-------------|----------------------------|
| NAMESPACE |      NAME      | TARGET PORT |            URL             |
|-----------|----------------|-------------|----------------------------|
| default   | springboot-app | http/8080   | http://192.168.49.2:30712  |
|-----------|----------------|-------------|----------------------------|
🎉  Opening service default/springboot-app in default browser...
👉  http://192.168.49.2:30712
```

Step 8: Output

The output is shown in the browser in the localhost url present qcewcwedcdwcdw

HOME FIND OWNERS VETERINARIANS ERROR

???welcome Aranganathan 79998wdd???